Risk and avoidance strategy for blocking mechanism of SDN-based security service

Minjae Byun*, Yongjun Lee*, Jin-Young Choi*
*Department of Information Security, Korea university, Seoul, Korea
minjae_byun@korea.ac.kr, yjlee@formal.korea.ac.kr, choi@formal.korea.ac.kr

***Abstract*—Software-Defined Network (SDN) is the dynamic network technology to address the issues of traditional networks. It provides centralized view of the whole network through decoupling the control planes and data planes of a network. Most SDN-based security services globally detect and block a malicious host based on IP address. However, the IP address is not verified during the forwarding process in most cases and SDN-based security service may block a normal host with forged IP address in the whole network, which means false-positive. In this paper, we introduce an attack scenario that uses forged packets to make the security service consider a victim host as an attacker so that block the victim. We also introduce cost-effective risk avoidance strategy.**

***Keywords*—SDN-based Security Services, risk analysis, IP forging, blocking mechanism, SDN attack**

I. Introduction

Software-Defined Network (SDN) is the dynamic network technology to address the issues of traditional networks. It provides centralized view of the whole network through decoupling the control planes and data planes of a network. The control planes are responsible for policy creation and its implementation and the data planes are responsible for packet forwarding [1]. This separation makes the network more programmable [2].

Recently, SDN-based security services like FRESCO [3], Avant-guard [4], Floodguard [5], Of-guard [6] and FlowRanger [7] were proposed. A centralized SDN-based security service can manage network resource and dynamically add or delete security rules [8]–[12]. Most SDN-based security services detect and block a malicious host based on IP address. It doesn't consider a flaw of Internet protocol that source IP address is not checked during the forwarding process [13] and the centralized management system blocks the IP address in the whole network [14]. As a result, malicious packets with forged IP address can cause a normal host to be blocked.

In this paper, we introduce an attack scenario that uses forged packets to make the security service consider a victim host as an attacker so that block the victim. An attacker sends malicious packets, whose source IP address is a victim's IP address, to SDN. The security service then detects these packets and considers the victim malicious. Consequently, the security service generates flow rules that blocks the victim and the victim is blocked globally in the network.

This research was sponsored by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2017M3C4A7083676).

```
/* Port numbering. Ports are numbered starting from 1. */
enum ofp_port_no {
    /* Maximum number of physical and logical switch ports. */
    OFPP_MAX = 0xffffff00,
    /* Reserved OpenFlow Port (fake output "ports"). */
    OFPP_UNSET = 0xfffffff7, /* Output port not set in action-set.
    used only in OXM_OF_ACTSET_OUTPUT. */
    OFPP_IN_PORT = 0xfffffff8, /* Send the packet out the input port.
    This reserved port must be explicitly used
    in order to send back out of the input
    port. */
    ...
};
```

Fig. 1. Port Structures in OpenFlow switch protocol

To prevent this attack, we also introduce cost-effective risk avoidance strategy. With our strategy, SDN-based security service blocks malicious IP address within the switch port from which the packet comes. This can prevent false positives from causing problems with availability of normal hosts. The rest of the paper is structured as follows. Section II presents the background of this paper. In section III, we explain IP-forged attack that can leads the loss of asset availability. Section IV gives risk avoidance strategy to prevent this attack. Finally, conclusion and future work are provided in Section V.

II. Background

A. OpenFlow structure

OpenFlow is the first and most widely used SDN standard interface. An OpenFlow controller is above a set of OpenFlow-enabled switches. It has a logic for the network's flow rule production, providing necessary flow rule updates to the switch [3]. For an OpenFlow switch, the data plane is programmable, where flows are dynamically specified within a flow table [3]. OpenFlow switch specification [11] defines the OpenFlow switch protocol used to configure flow table. Fig. 1 shows the Port Structures in the OpenFlow switch protocol. The field named OFPP_IN_PORT means the port where the packet enters the OpenFlow switch. We can monitor this field and execute predefined policies when it matches a certain value.

B. Detecting and Blocking mechanism of SDN-based security services

Some studies proposed SDN-based security services to mitigate various security challenges. The control plane of SDN makes centralized decisions based on the global view of the network. As a result, the SDN architecture supports reactive security management system to security policy alteration and

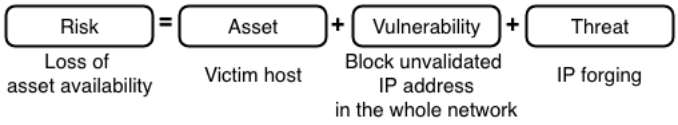

Fig. 2. Asset, Vulnerability, Threat and Risk

security service insertion [15]. A centralized SDN-based security service can manage network resource and dynamically add or delete security rules [8]–[12].

Most SDN-based security services detect and block malicious host as follows [8].

1) SDN-based security service centrally monitors the whole network.
2) During the monitoring, SDN-based security service analyzes the headers and contents of the packet.
3) If SDN-based security service regards the packet as malicious with suspicious patterns, SDN-based security service generates a security policy to block the corresponding IP address and sends the policy to the controller.
4) The SDN controller installs new rules from the security service (e.g., drop packets with the suspicious pattern) into switches.
5) The IP address from which the malicious packets came is blocked in the whole network.

III. IP Forging Attack

IP forging attack, which we introduce in this paper, is using the flaw of SDN-based security service. An attacker sends malicious packets, whose source IP address is a victim's IP address, to SDN. The security service then detects these packets and considers the victim malicious. Consequently, the security service generates flow rules that blocks the victim and the victim is blocked globally in the network.

In this section, we first analyze the risk of SDN-based security service. And then we define vulnerable component, attacker and victim with NVD's CVSS v3.0 [16]. Finally, we introduce IP forging attack scenario and evaluate the scenario with CVSS v3.0.

A. Risk of blocking mechanism of SDN-based security service

ISO/IEC 15408-2:2008 defines the content and presentation of the security functional requirements to be assessed in a security evaluation using ISO/IEC 15408 [17]. Based on ISO/IEC 15408-2:2008, risk occurs when asset, vulnerability and threat are combined. Fig.2 shows the relationship of asset, vulnerability, threat and risk.

- **Asset:** Assets are entities that someone places value upon. In our SDN environment, asset can be hosts. Impairment of the assets commonly includes loss of asset confidentiality, integrity and availability.
- **Vulnerability:** Vulnerability is weakness that can be exploited by a threat agent or an attacker. Most SDN-based security services have two flaws that can cause vulnerabilities.
  1) SDN-based security service uses Internet protocol that does not verify source IP address during the forwarding process.
  2) SDN-based security service blocks IP address in the whole network.
- **Threat:** A threat consists of an adverse action performed by a threat agent on an asset. An attacker can exploit vulnerabilities above with IP forging attack. The attacker uses forged packets to make the security service consider a victim host as the attacker so that block the victim.
- **Risk:** Due to the vulnerability that SDN-based security service centrally blocks IP address in the whole network, an attacker can do IP forging attack to a normal host. This lets the loss of asset availability.

B. Definition of vulnerable component

A vulnerable component defined in CVSS v3.0 represents characteristics of thing that is vulnerable [16].

A vulnerable component is the entire network in which the SDN-based security service runs. As described in II.B, SDN-based security service centrally monitors and controls network resources. Malicious packets are automatically detected by the centralized security service in the whole network, so an attacker doesn't have to send packets directly to it. Instead, the attacker can send packets in the network. The security service generates flow rules after detection and distributes them to switches. Consequently, the rules generated based on the detected packets will affect the whole network. In short, the attack will also affect the whole network. This characteristic is the key point of this attack and makes the attack surface wider.

A vulnerable component is the SDN-based security service that detects and blocks malicious actions. A principle of our attack is to make security service consider a victim as an attacker so that block the victim. Security services utilize the global view for detecting and blocking malicious actions. These kinds of security services can be our target.

C. Definition of attacker and victim

An attacker knows victims IP address. An attacker only needs to know the IP address of a victim because IP forging attack is network-based attack.

An attacker can send spoofed packets to SDN network. IP forging attack sends spoofed packets in the network. So first, an attacker makes spoofed packets. Second, an attacker should be able to send packets to SDN.

An attacker knows the actions to be detected by SDN-based security services. IP forging attack utilizes blocking mechanism in SDN-based security services and an attacker should know the mechanism or signature for detecting and blocking malicious actions.

A victim is in SDN-based security service area. IP forging attack utilizes the centralized SDN-based security service

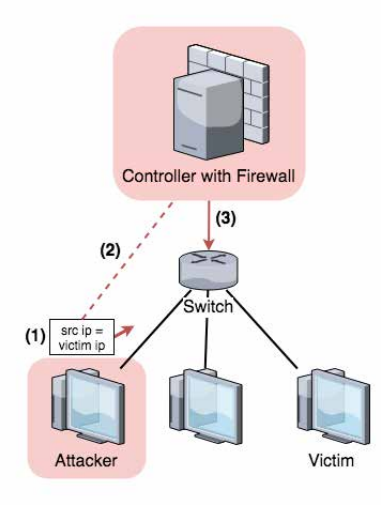

Fig. 3. IP Forging Attack scenario

mechanism. Therefore, to block a victim, a victim has to be within the area of the SDN-based security service.

D. IP Forging Attack Scenario

Fig. 3 shows the scenario of IP Forging attack as follows.

1) An attacker generates packets whose source IP address is a victim's IP address. The packets can contain various malicious payloads that should be detected by security services. Then the attacker sends packets to any host in the SDN.
2) When the attacker sends malicious packets to any host through SDN, the security service detects the packets using policies. The security service considers the victim as an attacker and generates a flow rule which blocks the victim.
3) The controller distributes the generated flow rule to switches. Each switch then updates the flow rule into its flow table.
4) As a result, the victim is blocked at every switch, which means it is blocked in the whole network.

E. Impact

CVSS(the Common Vulnerability Scoring System) provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity [16]. According to CVSS v3.0, the vulnerable component is the SDN-based security service and the impact component is the victim host. Evaluation details are as follows.

Attack Vector: IP forging attack is remotely exploitable, so Attack Vector is Network.

Attack Complexity: Attack Complexity is High because IP forging attack needs configuration settings to enable detection and blocking threats in the SDN-based security service. But unlike MiTM attack, IP forging attack can be done by simply one step.

Privileges Required: Privileges Required is None. The attacker only needs to send spoofed packets to SDN, so it is possible without privilege of the vulnerable component.

User Interaction: User Interaction is also None because IP forging attack doesn't require any user interaction.

Scope: When the victim is an important server or critical device, IP forging attack can affect many systems. So Scope is Changed.

Confidentiality Impact: Confidentiality Impact is None.

Integrity Impact: Integrity Impact is None.

Availability Impact: A IP forging attacked host is completely unavailable, so Availability Impact is High.

IV. Risk Avoidance Strategy

A. Risk avoidance strategy

In this section, we propose risk avoidance strategy to prevent IP forging attack. The attack described above occurs when the SDN-based security service blocks unverified IP address in the whole network. Source validation is a good way to prevent this attack but it needs complex deployment steps and high cost. So, we focus on the specification of the blocking range. We use OpenFlow protocol described in II.A to specify the blocking range.

Using OpenFlow protocol, we can figure out OFPP_IN_PORT, from which the malicious packets come, and set the specific blocking range. Applying our risk avoidance strategy, SDN-based security service blocks IP address when both IP address and port information meet the detection criteria. As OFSwitch drops malicious packets based on IP address and OFPP_IN_PORT, we can prevent IP forging attack from blocking a normal host.

B. Evaluation

We evaluate our strategy in terms of cost, correctness, and flexibility.

- **cost:** Our risk avoidance strategy using OpenFlow protocol only needs to add one blocking condition. It doesn't need any other algorithm or framework but check one more condition at OFSwitch. So, it can be implemented simply with low cost.
- **correctness:** Functional correctness refers to the input-output behavior of the algorithm (i.e., for each input it produces the expected output) [18]. Our strategy avoids risk and does not consider whether a packet has been forged. However, if an attacker tries IP forging attack, our strategy can perfectly prevent damage to an impact component, which means a victim host with the corresponding IP address. Our strategy also blocks an attacker attempting to attack using his own IP address. It can completely prevent the scenario introduced in III.D.
- **flexibility:** For networks, flexibility would refer to the ability to adapt the available network resources, such as flows or topology, to changes of design requirements, e.g., shorter latency budgets or different traffic distributions [19]. Some anti-IP-spoofing techniques in SDN drops forged packets through IP prefix [20], [21]. However, it

is inflexible and hard to cope with the situations such as topology dynamics and routing asymmetry [22]. Our approach can be applied to more flexible topology as it doesn't use IP prefix.

V. Conclusion

We propose the IP forging risk of SDN-based security service. SDN-based security service blocks unverified IP address in the whole network and this can cause the loss of asset availability. Avoidance strategy for this attack uses OpenFlow protocol to specify the blocking range. With this strategy, SDN-based security service blocks malicious IP address within the switch port from which the packet comes and we can prevent the loss of asset availability with cost-effective and simple way.

As future work, we will verify our risk avoidance strategy through experiments with SDN-based security services.

References

[1] P. Patel, Implementing software-defined network (SDN) based firewall, 2016.

[2] M. Rouse, Programmable Network (PN), 2013.

[3] S. W. Shin, P. Porras, V. Yegneswara, M. Fong, G. Gu, and M. Tyson, Fresco: Modular composable security services for software-defined networks, in 20th Annual Network Distributed System Security Symposium. NDSS, 2013.

[4] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, Avant-guard: Scalable and vigilant switch flow management in software-defined networks, in Proceedings of the 2013 ACM SIGSAC conference on Computer communications security. ACM, 2013, pp. 413424.

[5] H. Wang, L. Xu, and G. Gu, Floodguard: A dos attack prevention extension in software-defined networks, in 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE, 2015, pp. 239250.

[6] W. Haopei, X. Lei, and G. Guofei, Of-guard: a dos attack prevention extension in software-defined networks, USENIX Open Network Summit, 2014.

[7] L. Wei and C. Fung, Flowranger: A request prioritizing algorithm for controller dos attacks in software defined networks, in Communications (ICC), 2015 IEEE International Conference on. IEEE, 2015, pp. 5254 5259.

[8] J. Jeong, J. Seo, G. Cho, H. Kim, and J.-S. Park, A framework for security services based on software-defined networking, in Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on. IEEE, 2015, pp. 150153.

[9] R. I.-T. Y.3300, Framework of Software-Defined Networking, 2014.

[10] Open Networking Foundation, SDN Architecture, ONF TR-521, 2014.

[11] Open Networking Foundation, OpenFlow Switch Specification (Version 1.5.1), ONF TS-025, 2015.

[12] M. [27] Boucadair and C. Jacquenet, Software-defined networking: A perspective from within a service provider environment. No. RFC 7149. 2014.

[13] J. Wu, J. Bi, X. Li, G. Ren, K. Xu, and M. Williams, A source address validation architecture (sava) testbed and deployment experience, Tech. Rep., 2008.

[14] D. Jacobs, Addressing SDN security challenges means securing the controller, 2017.

[15] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, Security in software defined networks: A survey, IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 23172346, 2015.

[16] NVD, Common Vulnerability Scoring System v3.0: Specification Document

[17] ISO/IEC 15408-2:2008

[18] Douglas D., and Victor R. Basili. A Comparative Analysis of Functional Correctness. No. TR-921. MARYLAND UNIV COLLEGE PARK DEPT OF COMPUTER SCIENCE, 1980.

[19] Kellerer, Wolfgang, Arsany Basta, and Andreas Blenk. "Flexibility of Networks: a new measure for network design space analysis?." arXiv preprint arXiv:1512.03770 (2015).

[20] G. Yao and et al., Source address validation solution with OpenFlow/NOX architecture, in Proceedings of the 19th ICNP, 2011, pp. 712.

[21] S. Scott-Hayward and et al., A Survey of Security in Software Defined Networks, IEEE Communications Surveys Tutorials, vol. 18, no. 1, pp. 623654, 2016.

[22] Chen, Guolong, et al. "SAVSH: IP source address validation for SDN hybrid networks." Computers and Communication (ISCC), 2016 IEEE Symposium on. IEEE, 2016.

Minjae Byun received the two B.S. degrees in computer science and engineering and information security from Korea University, Seoul, South Korea, in 2018. She is currently pursuing the M.E. degree in information security at Graduate School of Information Security, Korea University, Seoul, South Korea. Her current research interests are in secure software engineering, formal methods, and cryptography.

Yongjun Lee received the two B.S degrees in Department of computer and information security and optical engineering from Sejong University, Seoul, South Korea, in 2018. He is currently pursuing the M.E. degree in information security at Graduate School of Information Security, Korea University, Seoul, South Korea. His current research interests are in formal methods, security vulnerability assessment, deep learning, and secure software engineering.

Jin-Young Choi received the M.S. degree from Drexel University, Philadelphia, PA, USA, in 1986, and the Ph.D. degree from the University of Pennsylvania, Philadelphia, PA, USA, in 1993. He is currently a Professor with the Graduate School of Information Security, Korea University, Seoul, South Korea. His current research interests are in real-time computing, formal methods, programming languages, process algebras, security, and secure software engineering.